US-PAT-NO: 6460082

DOCUMENT-IDENTIFIER: US 6460082 B1

TITLE: Management of service-oriented resources across heterogeneous media servers using homogenous service units and service signatures to configure the media servers

---------- KWIC ---------

Application Filing Date - AD (1):
19990617

Brief Summary Text - BSTX (7):
The task of leveraging the increased availability of widely distributed content and resources becomes very important with the proliferation of the next generation of the Internet, e.g., Internet2. There are a number of publications and patents in the area of QoS-driven resource management. Most of the work has been focused on either the network, as described in U.S. Pat. No. 5,388,097 issued Feb. 7, 1995 to Baugher, M. J. et al., and entitled "System and Method for Bandwidth Reservation for Multimedia Traffic in Communication Networks," and U.S. Pat. No. 5,581,703 issued Dec. 3, 1996 to Baugher, M. J. et al, and entitled "Method and Apparatus for **Reserving** System **Resources** to assure Quality of Service"; or, the operating system, such as described in the reference "An Architecture Towards Efficient OS Support for Distributed Multimedia", Proceedings of IS&T/SPIE Multimedia Computing and Networking Conference '96, San Jose, Calif., January 1996 by David K. Y. Yau and Simon S. Lam. With the proliferation of multimedia services on Internet, it was soon realized that while IP networks were able to provide a simple, best-effort delivery service, the IP protocol is not suited for use with new real-time applications, such as multimedia streaming, Virtual Reality applications, distributed supercomputing. As a result, new network protocols, such as Resource Reservation Setup Protocol (RSVP) (See, e.g., "The Grid: Blueprint for a New Computing Infrastructure," Edited by Ian Foster and Carl Kesselman, Chapter 19, pp. 379-503, Morgan Kauffman Publishers, 1999); Real Time Transport Protocol (RTP); Real Time Transport Control Protocol (RTCP) and others, were developed (See, e.g., William Stallings, "High-Speed Networks: TCP/IP and ATM Design Principles", Prentice Hall, 1997; and, I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS Control of Multimedia Applications based on RTP", Computer Communications, January 1996), enabling applications to request and negotiate network QoS parameters, such as bandwidth and latency. Deployment of those protocols on the current Internet has not been successful, firstly because it required upgrading all the non-RSVP routers and servers system software. Secondly, even if RSVP were deployed on the current Internet, very limited bandwidth and computing resources would still have been the bottleneck for successful deployment of real-time applications. The current Internet was built on the backbone, enabling cross-country communications on

relatively unclogged T3 (45 megabit per second). Proliferation of graphic pages, and streaming audio and video applications depleted those resources quite fast. Even worse, the rate of user's population growth is considerably higher than newly build network resources.

Brief Summary Text - BSTX (11):
A new breed of high performance applications such as remote surgery, robotics, tele-instrumentation, automated crisis response, digital libraries of satellite data, distance learning via multimedia supported Web sites, enhanced audio, and video, is emerging. However, to accommodate such high performance applications and their continuous media flows, it is not enough to increase or reserve network capacity. These new applications require end-to-end **resource** reservation and admission control, followed by co-ordination of distributed functions such as: (a) **resource** scheduling (e.g., CPU, disk, etc.) at the end-system(s), (b) packet scheduling and flow control in the network, and (c) **monitoring** of the delivered end-to-end quality of service. It is essential that quality of service is configurable, predictable and maintainable system-wide, including the end-system devices, communications subsystem, and networks. Furthermore, all end-to-end elements of distributed systems architecture must work in unison to achieve the desired application level behavior.

Brief Summary Text - BSTX (16):
For example, it would be desirable to determine commonality for the usage history of a particular multimedia content, e.g., bursts of requests within short time intervals, the proximity of origination addresses of requests, etc. In addition, the architectures described above do not allow for dynamic **monitoring** and recording of **resource** consumption for individual services as well as for groups of related services, with the purpose of calculating cost of service for individual clients.

Detailed Description Text - DETX (18):
As mentioned, a system administrator configures overall resources as local and global. The administrator is responsible for establishing the ratio of local to global resources for each server as well as to establishing **policies** relating to load limits for those resources. After configuring a partition as global storage, the global resource management takes over the control of this resource. Thus, a global storage bin represents a partition that can only be **reserved** by the global **resource** management provided by SCP. Note, that the system administrator or the server (500) itself, depending on a relevant **policy,** may re-claim the global resource in full or partially, by requesting its release from the SCP management. A native reservation management process (520) as shown in FIG. 5, is responsible for **monitoring resource** consumption, and determining the server willingness to accept new requests. It is understood that, the native process may discriminate between application requests for global and local content. Additionally, a request for placement of a global replica may be declined or accepted, as controlled by internal meta-resource **policies** governing the autonomy of the meta-resource with respect to admission control for example, one such **policy** could attempt to maximize the revenue associated with global resources and thus reject replicas expected to

have high cost or little revenue. Such a **policy** may also depend on request statistics monitored by the SCP.

Detailed Description Text - DETX (21):

For each type of service (i.e., capability) being provided by a meta-resource as indicated in a capabilities bank (130), a number of service units is pre-allocated. Thus, as shown in FIG. 4, ten (10) service units "A" may be committed to a local pool, while five (5) service units "A" may be committed to the local pool. Preferably, the pre-allocation is done independently for both the global and local pools. The administration of the meta-resource defines this number (herein referred to as the capacity of a capability) according to some criteria such as expected revenue or availability. For example, a meta-resource could be made to be from 0% to a 100% global (i.e., shareable with the global meta-system), as determined by the meta-resource administration. It should be noted that resources associated with fractional service units in a meta-resource might be allocated into a third pool referred to as an overflow pool (not shown). That is, after allocating both global and local service bins according to requirements or needs for provisioning services, there may be resources leftover. These remainder resources are managed as a overflow pool (733) as shown in FIG. 7. The overflow pool thus contains **resources that are not being reserved** for provisioning any service bins and thus, are free to be allocated by the meta-**resource** (server) as deemed necessary. For example, the overflow pool may be used to provide the resources needed to provide run-time resource compensation.

Detailed Description Text - DETX (28):

With further regard to compensation, as the possibility may exist that a **resource** envelope projected by a service unit may incorrectly estimate the **resource** requirements needed to provision the service object, existing interfaces are provided by the server operating system to permit **monitoring of resource** reservation exceptions. The resource management feedback unit (740) receives these exceptions and forwards these to the run-time compensation unit (735), which in turn computes the departure on the resulting resource envelope utilizing heuristics provided in a service unit heuristics database. Specifically, the **resource** management feedback module (740) is a software handler that maintains an association of individual **resource monitors** to a service unit and triggers a compensation of the **resource** envelope for a service unit during run-time. Once a service unit is allocated, individual **resource monitors** are started and associated with a common service unit. In case of an allocation exception or predicted under/over usage condition (against that predicted by the service unit's **resource** envelope) an exception may be fired by any one or more of these individual **resource monitors**. For example, if bandwidth is predicted to be low, the network I/O **monitor** (not shown) will signal such condition to the **resource** management feedback module which determines whether additional bandwidth needs to be allocated. To make such decision, the resource management feedback module (740) may rely on heuristics or **policies**. The service unit heuristics database (725) includes rules about how to operate over service unit as first class objects. For example, it knows/learns that for certain service units, allocating two of them at the same time really means allocating 2.times. the resources whereas for others it

means a worst case bound of, say, 1.4.times. the resources. This database additionally comprises data known to artisans in adaptive resource management such as bounds over resource adjustments, periodicity of resource adjustments, relative priorities of resource adjustments, etc.

Detailed Description Text - DETX (30):

FIG. 8(a) is a flow chart depicting in greater detail the process for handling a provisioning request (800). As shown in FIG. 8(a), the signaling adapter receives the provisioning request and then forwards any such request to the SUMM which then interfaces to the service unit database in order to retrieve and update resource envelopes (805). At step (810), the service unit signature for the particular requested service is compared with resources at a particular server. Specifically, when a request arrives at the meta-resource, it is necessary to determine whether the request can be serviced, i.e., if the meta-resource is capable, has the resources, is willing to, and has the necessary capability. All these decisions are abstracted by the service unit. Therefore, a determination is made at step (815) as to whether a service unit in a meta-resource is present indicating that the server is capable of provisioning such unit, i.e., that the necessary resources are present. The presence of a service unit provides the ability to determine the willingness of the server in accepting a request. If the service unit is not present, the request fails and the process ends without fulfillment of the request. If the service unit is present, then at step (820) a determination is made as to whether the meta-resource is willing to accept the request, i.e., if the server is willing to provide the media service when criteria such as price, current service unit utilization, and access controls, for example, are considered. Specifically, after a request arrives to the meta-resource, the meta-resource must decide whether to service the request or not. Such decision is supported by the meta-data in the resource. For example, the meta-resource (i.e., the server) determines whether the requests is associated with the right access controls (permissions) to use the service/storage bins being requested. Other criteria are price/cost admissibility. For example, the request may bound cost to $4.00 for example, whereas the meta-resource is willing to provide the service at $3.00. At step (825) the process will terminate if the request is not admissible, or, will continue otherwise. At step (835) any resource envelope adjustments are made and, at step (840), the adjusted service unit is allocated. For example, a service request may request a service unit (X, Y, Z) resource units of respective resources and is currently being serviced. A second request requests (X, Y, Z). For the adjustment step (835), a heuristics database look-up is performed and a determination made as to the form of the resulting resource allocation (f(X), g(Y), h(Z)) given the class of server (meta-resource). Once the resources are determined, any extra resources can be transferred to the overflow pool (e.g., for the duration associated for the provisioning of this request). This is accomplished during step (840) as well. Then, at step (850) the **resource monitors** are invoked by the operating system of the provisioning meta-**resource** (server) to **monitor** actual **resources** utilized in the provisioning of the requested service which is provided to the client as indicated at step (855). After provisioning of the service, the process ends at step (860) and returns to process more requests at step (865). Typically, the SUMM (FIG. 7) renders all its comparisons and determinations based on the corresponding resource envelope associated with a particular request and then requests the coordination and allocation of the service unit. However, the

coordination between the various resources associated with a particular service unit is provided by the coordinated resource management module (730). In turn, the coordinated resource management module interfaces with the resource management interfaces (750) provided by the operating system found on the meta-resource.

Detailed Description Text - DETX (31):

FIG. 8(b) is a flow chart depicting in greater detail the real-time **resource monitor** process thread invoked at step (850) of FIG. 8(a). Typically, this functionality is standard in most computer operating systems. For example, **resources** that may be **monitored** include virtual memory and page hits, stream I/O and buffer management, **CPU and CPU load** scheduling and priority handling, etc (See FIG. 7). As shown in FIG. 8(b), at step (875) the requirements departure is estimated, e.g., the number of I/O buffers needed to stream (e.g., 1 MB) and bandwidth. Techniques such as optimal smoothing of recorded and live video allow estimating reliably these values and determining under/over flow conditions. As it may be the case that the **resource monitor** (750) may not react on the first trigger, the **monitored** input data may be smoothed due to the nature of conventional operating systems. For example, an exponential smoother may be used. Trend assessment may additionally be performed as well. A critical threshold (that would be stored in the heuristics database) associated with each particular resource may be used to determine whether any departure is statistically significant, thus resulting in the generation of an exception (step 880). For example, correcting a 0.0001 departure is not a significant departure. Then, at step (880), a determination is made as to whether an exception has been detected. If an exception has not been detected, then the process returns at step (890). If an exception has been detected, then the resource allocation is compensated, as indicated at step (885) in the manner as described herein. Otherwise, the process returns at step (890).

Claims Text - CLTX (5):

5. The system as claimed in claim 3, wherein said server device further comprises: a) means for **monitoring resource** utilization when provisioning a media service; and b) means for compensating for differences between true resource utilization when providing a media service and its resource envelope as projected in its associated service unit.

Claims Text - CLTX (19):

19. The method as claimed in claim 17, further including the steps of: e) **monitoring** true **resource** utilization at a server device when providing said media service; and f) real-time compensating for differences between true resource utilization when providing said media service and its resource envelope as projected in its associated service unit.